



Implantando o barramento de serviços ERLANGMS no CPD/UFSM

Universidade Federal de Santa Maria

Centro de Processamento de Dados

Everton de Vargas Agilar (UnB / UFSM)

Jader Adiel (UFSM)





Plano

- Barramento de serviços ERLANGMS
 - Objetivo do projeto
 - Panorama geral sobre modernização
 - Histórico do projeto
 - Design da arquitetura
- ► Implantação do barramento na UFSM
 - O que está sendo feito
 - Desafios da migração do backend Delphi para Java
 - ► Plano de desenvolvimento da versão 2.0
 - Caso prático 1: Web services implementados para o NCC
 - Caso prático 2: TClis do SIE invocando web services Java





Barramento de serviços ERLANGMS





Objetivo do projeto

ERLANGMS

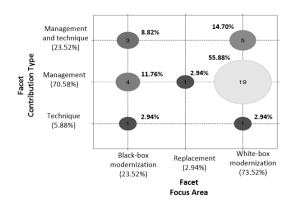
É uma abordagem desenvolvida na UnB para facilitar a modernização de sistemas para uma arquitetura orientada a serviços por meio de um barramento de serviços + SDK + processo de modernização e arquitetura documentado (SMSOC).





Panorama geral sobre modernização

Estudo realizado para identificar as estratégias de modernização







Histórico do projeto

2015 2016 2017 2018 2019

Primeira versão básica de um barramento REST em Erlang.

Foi escrito um SDK simples para desenvoyler servicos em Java usando os operadores filter. fields, sort e limit.

Estudo de caso para demonstrar o uso da REST API

O barramento recebeu suporte para HTTP 2.0 e HTTPS

aohsh

A interface da API foi melhorada com a possibilidade de realizar filtros mais complexos.

A capacidade de serialização foi ampliada para tratar melhor os tipos de

A API passou a suportar operadores de atributo.

Foco em desempenho com a Implementação de cache de queries usando named queries da JPA.

Implementação do ResultCache para cachear o resultado do webservice no harramento e evitar que requisições iquais seiam enviadas para o servidor de aplicação.

O SDK recebeu diversas melhorias de performance e estabilidade

Foco na segurança com suporte para dois protocolos: OAuth2 e I DAP v3

Foi incluído a possibilidade de gerar consultas com SQL nativo usando os operadores.

A SDK está sendo redesenhada para ser mais agnóstica e consequir trabalhar com Sprint e Java FF

O operador filter está sendo ampliado para suportar condições com and/or e precedência de condições.

Está sendo implementado suporte para API constraints para limitar o uso da API em cada endpoint.

A serialização de dados está sendo refinada para fazer a conversão de entidades e value objects (VO).





Histórico do projeto

Principais features em uso na UnB

Features

- ► HTTP/2, HTTP/1.1, HTTPS (Secure TLS Listener);
- ► LDAP v3 Proxy LDAP;
- ► HTTP Basic authentication;
- OAuth2 authentication;
- SDK para criar serviços em Java;
- ► Query Api filter, fields, sort, limit.
- ► Suporte para Dados Abertos.





Design da arquitetura

Design da arquitetura

- Multiplataforma e arquitetura modular;
- Serviços especificados em catálogos de serviços;
- Cluster de serviços para evitar ponto único de falhas;
- Suporta servi
 ços RESTful por contrato;
- ► Modelo de concorrência para serviços em Erlang: Actor Model;
- ► Baixa curva de aprendizado.





Implantação do barramento na UFSM





O que está sendo feito

O que já foi feito

- Implantação do barramento em partes, priorizando:
 - Query API
 - Subsistema de serialização;
 - Adaptar a interfaces do SDK para trabalhar de forma agnóstica com Java EE e Spring.
- ► Features visando a migração de backends SIE para Java:
 - ► Novo operador format para serializar dados como Dataset;
 - Novo subsistema provider no SDK para permitir seu uso independente da camada de negócio;
 - Provider para trabalhar com meta-queries do Delphi;
 - Cliente REST implementado no Delphi.





O que ainda precisa ser feito x prioridade

O que ainda precisa ser feito

- Tornar a camada de fachada mais simples Moderado;
- ► Implementar o subsistema de autenticação OAuth2 (ou integrar o barramento que já possui) — Alto;
- Analisar os principais métodos do TCliBusiness e prover um suporte adequado no SDK para facilitar a migração – Alto;
- Permitir gerar uma consulta por meio da Query API para buscar dados de um atributo lista de um objeto principal – Baixo;
- ▶ Permitir encadear funções nos atributos das entidades com o operador % para modificar a forma como o dado é lido da fonte de dados – Baixo.





12/23

Desafios da migração do backend Delphi para Java

Desafios da migração do backend Delphi para Java

- Basicamente, qualquer parte do Delphi é um desafio enorme levando em consideração o custo de tempo e complexidade do código;
- A única excessão será as meta-queries que serja bem fácil com o provider de meta-query do SDK;
- Os métodos das TClis possuem muito herança de código com muitos métodos virtuais que dificulta extrair a lógica de negócio sem precisar refazer tudo no Java;
- ► Para mapear os dados em Dataset, é necessário especificar o schema dos Dados:





Principais Resultados do Trabalho





Desenvolvimento de serviços com o SDK







Autenticação de usuários com o proxy LDAP







Autenticação e autorização de serviços REST com OAuth2







Trabalhos futuros

Trabalhos futuros

- ► Portal Api Management;
- ► JSON Schema Draft 4;
- ► SDK .Net.





Perguntas?





Perguntas

- ► (QP1) Por que Erlang/OTP?
- ► (QP2) Quem usa Erlang/OTP?
- ► (QP3) Por que o desenvolvimento de um novo barramento em vez de utilizar um existente?
- ► (QP4) Por que não implementar web services usando somente Java e seus frameworks?





(QP1) Por que Erlang/OTP?

- ► Suporta aplicações distribuídas e tolerantes a falhas a serem executadas em um ambiente de tempo real e ininterrupto;
- Possui um ambiente de execução que realmente facilita o desenvolvimento de software de rede e de clusters de serviços;
- Possui um modelo de arquitetura baseado em atores (Actor Model) onde os processos podem se comunicar apenas por mensagens.





(QP2) Quem usa Erlang/OTP?

- ► Facebook (Backend do chat 100 milhões de usuários ativos);
- Whatapps (Servidores de mensagens 2 milhões de usuários/servidor);
- Yahoo (Delicious − 5 milhões de usuários e mais de 150 milhões de bookmarks);
- Amazon SimpleDB, o serviço de dados do Amazon EC2;
- GitHub (Backend milhares de transações concorrentes);
- ► T-Mobile nos seus sistemas de SMS e autenticação;
- ► Motorola, CouchDB, RabbitMQ, Ejabbed, entre outros.





(QP3) Por que um novo barramento?

- Obter domínio sobre as tecnologias e o design RESTful;
- Barramento de serviços são produtos caros e complexos;
- ▶ Disponibilizar um produto simples (contém somente 10 mil linhas de código atualmente).





(QP4) Por que não implementar web services em Java com frameworks?

- ► É uma opção, mas exige ferramentas para gerenciar;
- ► Não é orientado a contrato de serviços;
- Não é independente de linguagem de programação;
- ► Não é escalável apenas usando os frameworks.





Obrigado!